



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/994,525	11/26/2001	Yehoshua Hershberg	50325-0621	8525

29989 7590 11/18/2004

HICKMAN PALERMO TRUONG & BECKER, LLP
1600 WILLOW STREET
SAN JOSE, CA 95125

EXAMINER	
GORDON, CARLENE MICHELLE	
ART UNIT	PAPER NUMBER

2124

DATE MAILED: 11/18/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Best Available Copy

Office Action Summary

Application No.

09/994,525

Applicant(s)

HERSHBERG ET AL.

Examiner

Carlene Gordon

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 November 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-23 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-23 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 11/26/04.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

DETAILED ACTION

1. This application is responsive application filed on November 26, 2001.

Claims 1- 23 are pending in the application.

Specification

2. The disclosure is objected to because of the following informalities:
 - a) The numbering of paragraphs on pg. 4 of the specification [00010] and [0010].
 - b) In paragraph [0048] pg. 16, "13" should be replaced by "12".
 - c) In paragraph [0048] pg. 17, reference to "line 7.5" of Table 2 appears inappropriate.
 - d) Numerous errors in relation to the several "Table"(s) of the specification were encountered throughout the specification, including line numbering of Table 6 on pg. 26 and references to the line numbers.

Appropriate correction is required.

3. The use of the trademark "JAVA" has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 4 and 18 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

6. Claim 4 recites the limitation "the data exchange format as described by **the second data**" in line 3 of claim 4. There is insufficient antecedent basis for this limitation in the claim.

For the purpose of examining, the Office will interpret "the second data" as the "**the third data**" to have proper antecedent basis.

7. Claim 18 recites the limitation "building the application based on the source code and at least one of the module for marshaling data and the module for de-marshaling data". The limitation "the application" takes antecedent basis from "an application having source code" of claim 17. It is unclear as to whether the applicant intends to claim the building of a new application, also, it is unclear how "the application" of claim 18 can be built based on source code stated to have already been included in the "application" of claim 17. Furthermore it is unclear how "the application" of claim 18 can be built based on "at least one of

Art Unit: 2124

the module for... and the module for" when these modules are built based on "first data" already included in the source code.

For proper prosecution of the limitation:

building the application based on the source code and at least one of the module for marshaling data and the module for de-marshaling data,

this limitation of claim 18 will be interpreted as:

building an application based on the source code and at least one of the module for marshaling data and the module for de-marshaling data.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1-3, 7-8, 10-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sundareson (USPN 6,675,370), hereafter "**Sundaresan**", and in further view of Flanagan (Java In a Nutshell), hereafter "**Flanagan**".

9. As to claim 1:

Sundaresan discloses receiving, from a source code file, comment data (col. 1 lines 56-60 "parses... comments")

Art Unit: 2124

first data indicating a parameter of the data exchange format (col. 3 lines 35-42 "custom Javadoc tag..." "XML structure"),

a source code processor of the source language (col. 1 lines 56-60 "Java compiler");

receiving from the source code file second data indicating a statement that defines a class of data objects in the source language (col. 1 lines 56-58, "parses... declarations... in Java source files... classes"; col. 4 lines 45-48 "productions in... classes"); and

automatically generating (col. 4 lines 21-25 "automatically generated"), based on the first data and the second data (col. 4 lines 15-25 "source code is... processed" – Wherein the 1st and 2nd data is included.), third data that describes the data exchange format (col. 4 lines 21-25 "HTML documentation for each class"; lines 49-51 "HTML corresponding to a production"; col. 3 lines 35-42 "custom Javadoc tag, @production... class" "XML structure" – Wherein HTML documentation comprises the third data.).

Sundaresan does not explicitly disclose the comment data includes the first data, the comment data being ignored, nor does Sundaresan explicitly disclose the second data associated with the comment data.

Flanagan teaches an analogous art of utilizing a Javadoc tool with Java source code. Flanagan shows a first data, similar to the data of Sundaresan, known to one of ordinary skill in the art at the time of the applicant's invention, to be included in the comment data (pg. 233 under **Java Documentation Comment Syntax** paragraph 2 "doc-comment are embedded within a Java

Art Unit: 2124

comment”; pg. 254 under **Description** paragraph 5 “A doc comment... include tag values... javadoc”; pg. 233 under **Java Documentation Comment Syntax** paragraph 3 “doc-comment... javadoc tag”). It would have been obvious to one of ordinary skill in the art at the time of the applicant’s invention to insert within comment data, the first data of Sundaresan because the first data is included in the comment data as shown by Flanagan to be parsed by Javadoc separately of the declarations of the source language. The data contained within the comment data is not to be interpreted as a part of the source language to be compiled by the Java compiler.

Flanagan teaches it known to one of ordinary skill in the art that comment data is ignored by a source code processor of the source language (pg. 20 under **Comments** “comment out... code”; pg. 233 under **Java Documentation Comment Syntax** paragraph 1 “comments are not treated by compiler”) of the source language (pg. 233 “Java language”). It would have been obvious to one of ordinary skill in the art that the comment data is being ignored in the context of Sundaresan’s invention because comment data is data, intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language to follow that must be compiled by the source code processor.

Flanagan teaches it known to one of ordinary skill in the art that a second data, as the second data of Sundaresan, to be associated with the comment data (pg. 233 “Doc-comments should immediately precede the declaration of the class... they are associated with.”). It would have been obvious to one of ordinary skill in the art at the time of the applicant’s invention that the second

Art Unit: 2124

data is associated with the comment data because comment data is intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language that follow.

10. As to claim 2:

Rejection of claim 1 is incorporated and further Sundaresan discloses generating, based on the first data and the second data (see claim 1), a module to convert a data object of the class of data objects into a data item (col. 4 lines 21-25 "HTML documentation for each class is generated" col. 4 lines 45-48 "productions in... classes" – Interpreted as software implemented steps.) of the data exchange format as described by the third data (see claim 1).

11. As to claim 3:

Rejection of claim 1 is incorporated and further Sundaresan discloses generating, based on the third data (see claim 1), a module to convert a data object of the class of data objects into a data item (col. 4 lines 21-25 "HTML documentation for each class is generated" col. 4 lines 45-48 "productions in... classes" – Interpreted as software implemented steps.) of the data exchange format as described by the third data (see claim 1).

12. As to claim 7:

Rejection of claim 1 is incorporated, and further Sundaresan discloses

Art Unit: 2124

wherein the third data is formatted according to a symbolic markup language (col. 3 line 35 – col. 4 line 1 “value associated with this tag is a XML structure”; “XML structure defined by this DTD” – Wherein XML is a symbolic markup language).

13. As to claim 8:

Rejection of claim 1 is incorporated and further Sundaresan discloses the third data formatted according to extensible markup language (XML) (col. 3 line 35 – col. 4 line 1 “value associated with this tag is a XML structure”; “XML structure defined by this DTD”).

14. As to claim 10:

Rejection of claim 1 is incorporated and further Sundaresan discloses wherein the third data is one or more statements in an XML document type definition (DTD) document (col. 3 lines 35-47; col. 4 lines 1-10, and lines 37-48 “a DTD defines a validly formed XML production” “creates the HMTL corresponding to the production”).

15. As to claim 11:

Rejection of claim 1 is incorporated and further Sundaresan discloses wherein the third data is one or more statements in an XML document type definition (DTD) document (col. 3 lines 35-47; col. 4 lines 1-10, and lines 37-48 “a DTD defines a validly formed XML production” “creates the HMTL corresponding

Art Unit: 2124

to the production”), and wherein the parameter is at least one of a root element associated with an entire DTD document, an element and an attribute of an element (col. 3 line 35 – col. 4 line 10 “XML structure... hierarchical tag structure” “XML structure defined by this DTD”).

16. As to claim 12:

Rejection of claim 1 is incorporated and further Sundaresan discloses wherein the third data is one or more statements in an XML document type definition (DTD) document (col. 3 lines 1-47; col. 4 lines 1-48 “a DTD defines a validly formed XML production” “creates the HMTL corresponding to the production”), and wherein the first data includes one or more properties of the parameter (col. 3 line 35 – col. 4 line 10 “tag occurrence in a program is a (tag name, tag value) pair” “the value associated...”).

17. As to claim 13:

Rejection of claim 1 is incorporated and further Sundaresan discloses wherein the source language is Java (col. 2 line 63 – col. 3 line 28 “Java”).

18. As to claim 14:

Rejection of claim 1 is incorporated and further Sundaresan discloses wherein the source language is Java (col. 2 line 63 – col. 3 line 28 “Java”), and wherein the first data includes a tag for an automated Java documentation system (col. 3 lines 10-42 “Javadoc tag”).

19. As to claim 15:

Rejection of claim 1 is incorporated and further Sundaresan discloses wherein the source language is Java (col. 2 line 63 – col. 3 line 28 “Java”), wherein the first data includes a tag for an automated Java documentation system (col. 3 lines 10-42 “Javadoc tag”), and wherein the tag is a user-defined tag for the Java documentation system (col. 3 lines 10-15 “Javadoc... allows user to define... own tags” “This extensibility is utilized...”).

20. As to claim 16:

Rejection of claim 1 is incorporated and further Sundaresan discloses wherein the source language is Java (col. 2 line 63 – col. 3 line 28 “Java”), wherein the first data includes a tag for an automated Java documentation system (col. 3 lines 10-42 “Javadoc tag”), wherein the tag is a user-defined tag for the Java documentation system (col. 3 lines 10-15 “Javadoc... allows user to define... own tags” “This extensibility is utilized...”), and wherein said step of generating the third data is performed by a user-defined routine invoked by the automated Java documentation system in response to the tag (Figure 1; col. 4 lines 10-25 “standard Javadoc doclet is extended by defining 104 a custom tag.. and creating 106 an extended doclet to process this tag” “user creating” “source code is processed by Javadoc processor using extended doclet” “automatically generated HTML documentation for each class”).

Art Unit: 2124

21. As to claim 17:

Sundaresan discloses inserting, into the source code (col. 3 and col. 4 lines 15-20 "source programs are created which include the... tags")

comment data (Fig. 1; col. 1 lines 56-59 "Javadoc parses... comments")

a processor for the particular language (col. 1 lines 56-60 "Java compiler"),

first data including a tag and a parameter of a data exchange format (col. 3 lines 29-44 "tag occurrence in a program is a (tag name, tag value) pair" "value associated... is an XML structure"); and

causing a processor to produce second data for configuring the data exchange format based at least in part on the first data (Fig. 1 "create extended doclet" col. 4 lines 15-25 "Javadoc doclet is extended by defining 104 a custom tag" "source code is processed by Javadoc processor" "generated HTML documentation for each class").

Sundaresan does not explicitly disclose the first data is inserted within the comment data, nor comment data being ignored.

Flanagan teaches an analogous art of utilizing a Javadoc tool with Java source code. Flanagan shows a first data, similar to the data of Sundaresan, known to one of ordinary skill in the art at the time of the applicant's invention, inserted in the comment data (pg. 233 under **Java Documentation Comment Syntax** paragraph 2 "doc-comment are embedded within a Java comment"; pg. 254 under **Description** paragraph 5 "A doc comment... include tag values... javadoc"; pg. 233 under **Java Documentation Comment Syntax** paragraph 3

Art Unit: 2124

"doc-comment... javadoc tag"). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to insert within comment data, the first data of Sundaresan because the first data is included in the comment data as shown by Flanagan to be parsed by Javadoc separately of the declarations of the source language. The data contained within the comment data is not to be interpreted as a part of the source language to be compiled by the Java compiler.

Flanagan teaches it known to those of ordinary skill in the art that comment data is ignored by a source code processor of the source language (pg. 20 under **Comments** "comment out... code"; pg. 233 under **Java Documentation Comment Syntax** paragraph 1 "comments are not treated by compiler") of the source language (pg. 233 "Java language"). It would have been obvious to one of ordinary skill in the art that the comment data is being ignored in the context of Sundaresan's invention because comment data is data, intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language to follow that must be compiled by the source code processor.

22. As to claim 18:

Rejection of claim 17 is incorporated and further Sundaresan discloses causing the processor to produce at least one of a module for marshaling data objects into a data item in the data exchange format as configured by the second data (Fig. 1 steps 110 and 112; col. 4 lines 10-25 "Javadoc processor" col. 4

Art Unit: 2124

lines 21-25 "HTML documentation for each class is generated" col. 4 lines 45-48 "productions in... classes" – Module is interpreted as software implemented steps) and a module for de-marshaling data objects from a data item in the data exchange format as configured by the second data and

building an application based on the source code and at least one of the module for marshaling data and the module for de-marshaling data (Fig. 3b; col. 4 lines 10-25 "source code is processed" "generated HTML documentation" – Wherein the module is interpreted as the software implemented steps of generating HTML documentation).

23. As to claim 19:

Rejection of claim 18 is incorporated and further Sundaresan discloses wherein the particular language is the Java language (col. 2 line 63 – col. 3 line 28 "Java"), wherein the tag is a user-defined tag of an automated Java documentation system (col. 3 lines 10-42 "Javadoc tag"; col. 3 lines 10-15 "Javadoc... allows user to define... own tags" "This extensibility is utilized..."); and wherein said step of causing a processor to produce the second data further comprises providing a routine, invoked by the automated Java documentation system in response to the tag, to produce the second data (Figure 1 shows automated Java documentation system in response to the tag; col. 4 lines 10-25 "standard Javadoc doclet is extended by defining 104 a custom tag.. and creating 106 an extended doclet to process this tag" "user creating" "source code is

Art Unit: 2124

processed by Javadoc processor using extended doclet” “automatically generated HTML documentation for each class”).

24. As to claim 20:

Rejection of claim 18 is incorporated and further Sundaresan discloses wherein the particular language is the Java language (col. 2 line 63 – col. 3 line 28 “Java”); wherein the tag is a user-defined tag of an automated Java documentation system (col. 3 lines 10-42 “Javadoc tag”; col. 3 lines 10-15 “Javadoc... allows user to define... own tags” “This extensibility is utilized...”); and wherein said step of causing a processor to produce at least one of the module for marshaling and the module for de-marshaling (col. 4 lines 20-25 “source code is processed” “generated HTML documentation” –Wherein the module is interpreted as the software implemented steps of generating HTML documentation) further comprises providing a routine, invoked by the automated Java documentation system in response to the tag (col. 4 lines 20-25 “by processor using extended doclet”; col. 4 lines 10-15 “Fig. 1 ... doclet is extended by defining 104 a custom tag”), to produce at least one of a Java module for marshaling and a Java module for de-marshaling (col. 4 lines 20-25 “HTML documentation for each class is generated” – see above).

25. As to claim 21:

Sundaresan discloses receiving, from a particular file that includes source

Art Unit: 2124

code, comment data (col. 1 lines 56-60 “parses... comments”) including first data indicating a parameter of the data exchange format (col. 3 lines 35-42 “custom Javadoc tag...” “XML structure” – Wherein the first data is inherently included in the comment data), wherein the comment data is ignored (It is inherent that comment data is ignored) by a source code processor of the particular language (col. 1 lines 56-60 “Java compiler”);

receiving from the particular file second data, associated with the comment data, indicating a statement that defines a class of data objects in the particular language (col. 1 lines 56-58, “parses... declarations... in Java source files... classes”; col. 4 lines 45-48 “productions in... classes” – It is inherent that the declarations are associated with the comment data”); and

generating (col. 4 lines 21-25 “generated”), based on the first data and the second data (col. 4 lines 15-25 “source code is... processed” – Wherein the 1st and 2nd data is included.), third data for configuring the data exchange format (col. 4 lines 1-25 “HTML documentation for each class”; lines 49-51 “HTML corresponding to a production”; col. 3 lines 35-47 “custom Javadoc tag, @production... class” “XML structure” – Wherein HTML documentation comprises the third data.).

Sundaresan does not explicitly disclose the comment data includes the first data, the comment data being ignored, nor does Sundaresan explicitly disclose the second data associated with the comment data.

Flanagan teaches an analogous art of utilizing a Javadoc tool with Java source code. Flanagan shows a first data, similar to the data of Sundaresan,

Art Unit: 2124

known to one of ordinary skill in the art at the time of the applicant's invention, to be included in the comment data (pg. 233 under **Java Documentation Comment Syntax** paragraph 2 "doc-comment are embedded within a Java comment"; pg. 254 under **Description** paragraph 5 "A doc comment... include tag values... javadoc"; pg. 233 under **Java Documentation Comment Syntax** paragraph 3 "doc-comment... javadoc tag"). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to insert within comment data, the first data of Sundaresan because the first data is included in the comment data as shown by Flanagan to be parsed by Javadoc separately of the declarations of the source language. The data contained within the comment data is not to be interpreted as a part of the source language to be compiled by the Java compiler.

Flanagan teaches it known to those of ordinary skill in the art that comment data is ignored by a source code processor of the source language (pg. 20 under **Comments** "comment out... code"; pg. 233 under **Java Documentation Comment Syntax** paragraph 1 "comments are not treated by compiler") of the source language (pg. 233 "Java language"). It would have been obvious to one of ordinary skill in the art that the comment data is being ignored in the context of Sundaresan's invention because comment data is data, intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language to follow that must be compiled by the source code processor.

Flanagan teaches it known to one of ordinary skill in the art that a second data, as the second data of Sundaresan, to be associated with the comment data (pg. 233 "Doc-comments should immediately precede the declaration of the class... they are associated with."). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention that the second data is associated with the comment data because comment data is intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language that follow.

26. As to claim 22:

Sundaresan discloses means for receiving, from a particular file that includes source code, comment data (col. 1 lines 56-60 "Javadoc... parses... comments") including first data indicating a parameter of the data exchange format (col. 3 lines 35-42 "custom Javadoc tag..." "XML structure" – Wherein the first data is inherently included in the comment data), wherein the comment data is ignored (It is inherent that comment data is ignored) by a source code processor of the particular language (col. 1 lines 56-60 "Java compiler");

means for receiving from the particular file second data, associated with the comment data, indicating a statement that defines a class of data objects in the particular language (col. 1 lines 56-58, "Javadoc... parses... declarations... in Java source files... classes"; col. 4 lines 45-48 "productions in... classes" – It is inherent that the declarations are associated with the comment data"); and

Art Unit: 2124

means for generating (col. 4 lines 21-25 "automatically generated"), based on the first data and the second data (col. 4 lines 15-25 "source code is... processed... by... processor" – Wherein the 1st and 2nd data is included.), third data for configuring the data exchange format (col. 4 lines 21-25 "HTML documentation for each class"; lines 49-51 "HTML corresponding to a production"; col. 3 lines 35-47 "custom Javadoc tag, @production... class" "XML structure" – Wherein HTML documentation comprises the third data.).

Sundaresan does not explicitly disclose the comment data includes the first data, the comment data being ignored, nor does Sundaresan explicitly disclose the second data associated with the comment data.

Flanagan teaches an analogous art of utilizing a Javadoc tool with Java source code. Flanagan shows a first data, similar to the data of Sundaresan, known to one of ordinary skill in the art at the time of the applicant's invention, to be included in the comment data (pg. 233 under **Java Documentation Comment Syntax** paragraph 2 "doc-comment are embedded within a Java comment"; pg. 254 under **Description** paragraph 5 "A doc comment... include tag values... javadoc"; pg. 233 under **Java Documentation Comment Syntax** paragraph 3 "doc-comment... javadoc tag"). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to insert within comment data, the first data of Sundaresan because the first data is included in the comment data as shown by Flanagan to be parsed by Javadoc separately of the declarations of the source language. The data contained within the comment

Art Unit: 2124

data is not to be interpreted as a part of the source language to be compiled by the Java compiler.

Flanagan teaches it known to those of ordinary skill in the art that comment data is ignored by a source code processor of the source language (pg. 20 under **Comments** "comment out... code"; pg. 233 under **Java Documentation Comment Syntax** paragraph 1 "comments are not treated by compiler") of the source language (pg. 233 "Java language"). It would have been obvious to one of ordinary skill in the art that the comment data is being ignored in the context of Sundaresan's invention because comment data is data, intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language to follow that must be compiled by the source code processor.

Flanagan teaches it known to one of ordinary skill in the art that a second data, as the second data of Sundaresan, to be associated with the comment data (pg. 233 "Doc-comments should immediately precede the declaration of the class... they are associated with."). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention that the second data is associated with the comment data because comment data is intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language that follow.

27. As to claim 23:

Sundaresan discloses a processor; one or more stored sequences of

Art Unit: 2124

instructions which, when executed by the processor, cause the processor to carry out the steps of (Fig. 1; cols 1-5 "processor"):

receiving, from a particular file that includes source code, comment data (col. 1 lines 56-60 "parses... comments") including first data indicating a parameter of the data exchange format (col. 3 lines 35-42 "custom Javadoc tag..." "XML structure" – Wherein the first data is inherently included in the comment data), wherein the comment data is ignored (It is inherent that comment data is ignored) by a source code processor of the particular language (col. 1 lines 56-60 "Java compiler");

receiving from the particular file second data, associated with the comment data, indicating a statement that defines a class of data objects in the particular language (col. 1 lines 56-58, "parses... declarations... in Java source files... classes"; col. 4 lines 45-48 "productions in... classes" – It is inherent that the declarations are associated with the comment data"); and

generating (col. 4 lines 21-25 "automatically generated"), based on the first data and the second data (col. 4 lines 15-25 "source code is... processed" – Wherein the 1st and 2nd data is included.), third data for configuring the data exchange format (col. 4 lines 21-25 "HTML documentation for each class"; lines 49-51 "HTML corresponding to a production"; col. 3 lines 35-47 "custom Javadoc tag, @production... class" "XML structure" – Wherein HTML documentation comprises the third data.).

Sundaresan does not explicitly disclose the comment data includes the first data, the comment data being ignored, nor does Sundaresan explicitly disclose the second data associated with the comment data.

Flanagan teaches an analogous art of utilizing a Javadoc tool with Java source code. Flanagan shows a first data, similar to the data of Sundaresan, known to one of ordinary skill in the art at the time of the applicant's invention, to be included in the comment data (pg. 233 under **Java Documentation Comment Syntax** paragraph 2 "doc-comment are embedded within a Java comment"; pg. 254 under **Description** paragraph 5 "A doc comment... include tag values... javadoc"; pg. 233 under **Java Documentation Comment Syntax** paragraph 3 "doc-comment... javadoc tag"). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to insert within comment data, the first data of Sundaresan because the first data is included in the comment data as shown by Flanagan to be parsed by Javadoc separately of the declarations of the source language. The data contained within the comment data is not to be interpreted as a part of the source language to be compiled by the Java compiler.

Flanagan teaches it known to one of ordinary skill in the art that comment data is ignored by a source code processor of the source language (pg. 20 under **Comments** "comment out... code"; pg. 233 under **Java Documentation Comment Syntax** paragraph 1 "comments are not treated by compiler") of the source language (pg. 233 "Java language"). It would have been obvious to one

Art Unit: 2124

of ordinary skill in the art that the comment data is being ignored in the context of Sundaresan's invention because comment data is data, intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language to follow that must be compiled by the source code processor.

Flanagan teaches it known to one of ordinary skill in the art that a second data, as the second data of Sundaresan, to be associated with the comment data (pg. 233 "Doc-comments should immediately precede the declaration of the class... they are associated with."). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention that the second data is associated with the comment data because comment data is intended for a viewer of the source code file, that describes the classes, methods, etc. of the source language that follow.

Claim Rejections - 35 USC § 103

28. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

29. Claims 6 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sundaresan and Flanagan as applied to claim 1 above, and further in view of "The XML Handbook" by Goldfarb et al., hereafter "**Goldfarb**".

Art Unit: 2124

30. As to claim 6:

Rejection of claim 1 is incorporated and further Sundaresan does not explicitly disclose wherein the third data is formatted according to a database query language.

Sundaresan discloses wherein the third data is one or more statements in an XML document type definition (DTD) document (col. 3 lines 35-47; col. 4 lines 1-10, and lines 37-48 "a DTD defines a validly formed XML production" "creates the HTML corresponding to the production").

However, Goldfard discloses an XML schema document, wherein data is formatted according to a database query language, as an alternate embodiment of a XML DTD (pg. 13 paragraph 4 "DTD's are schemas for documents.") wherein data is formatted according to a database query language.

At the time of the applicant's invention, it would have been obvious to use a XML schema document of Goldfard, wherein data is formatted according to a database query language, for defining a structure as Sundaresan teaches using a XML DTD. The motivation would have been because a XML schema document affords writing the set of constraints on a structure in a database (pg. 13 paragraph 4). The use of a XML schema document expands the extent of Sundaresan's invention to used with data in databases.

31. As to claim 9:

Art Unit: 2124

Rejection of claim 1 is incorporated and further Sundaresan does not explicitly disclose wherein the third data comprises one or more statements in an XML schema document.

Sundaresan discloses wherein the third data is one or more statements in an XML document type definition (DTD) document (col. 3 lines 35-47; col. 4 lines 1-10, and lines 37-48 "a DTD defines a validly formed XML production" "creates the HTML corresponding to the production").

However, Goldfard discloses an XML schema document as an alternate embodiment of a XML DTD (pg. 13 paragraph 4 "DTD's are schemas for documents.").

At the time of the applicant's invention, it would have been obvious to use a XML schema document of Goldfard for defining a structure as Sundaresan teaches using a XML DTD. The motivation would have been because a XML schema document affords writing the set of constraints on a structure in a database. The use of a XML schema document expands the extent of Sundaresan's invention to used with data in databases.

32. Claims 4-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sundaresan and Flanagan as applied to claim 1 above, and further in view of Tuatini (US Patent Application Publication No. 2001/0054172 A1), hereafter **"Tuatini"**.

33. As to claim 4:

Art Unit: 2124

Rejection of claim 1 is incorporated and further Sundaresan does not explicitly disclose generating, based on the first data and the second data, a module to derive a data object of the class of data objects from a data item of the data exchange format as described by the third data.

Sundaresan does disclose generating, based on the first data and the second data (see claim 1), a module to convert a data object of the class of data objects into a data item (col. 4 lines 21-25 "HTML documentation for each class is generated" col. 4 lines 45-48 "productions in... classes" – Interpreted as software implemented steps.) of the data exchange format as described by the third data (see claim 1). Also, Sundaresan discloses other functionally equivalent output formats, which include cross-linking... such as in XML (col. 5 lines 32-42).

However, Tuatini teaches the generation of serialization code (pg. 1 paragraph [0010] "XML serialization code") using XML DTD. This serialization code includes de-serialization code to derive a data object of a class of data objects from a data item of the data exchange format (pg. 2 paragraph [0012] "deserialization code maps the data of the XML document to the object" "outputs an object that is an instance of a class").

At the time the invention was made, one of ordinary skill in the art would have been motivated to combine the analogous art of using Java and markup languages for data exchange of Sundaresan and Tuatini to generate the de-serialization code of Tuatini as a module to derive a data object of the class of data objects from a data item of the data exchange format. The motivation would have been to provide flexibility needed by a group to provide applications tailored

Art Unit: 2124

to their customers as suggested by Tuatini on pg. 1 paragraph [0004], as the de-serialization code provides a means for this flexibility.

34. As to claim 5:

Rejection of claim 1 is incorporated and further Sundaresan does not explicitly disclose generating, based on the third data, a module to derive a data object of the class of data objects from a data item of the data exchange format as described by the third data.

Sundaresan does disclose generating, based on the third data (see claim 1), a module to convert a data object of the class of data objects into a data item (col. 4 lines 21-25 "HTML documentation for each class is generated" col. 4 lines 45-48 "productions in... classes" – Interpreted as software implemented steps.) of the data exchange format as described by the third data (see claim 1). Also, Sundaresan discloses other functionally equivalent output formats, which include cross-linking... such as in XML (col. 5 lines 32-42).

However, Tuatini teaches the generation of serialization code (pg. 1 paragraph [0010] "XML serialization code") using XML DTD. This serialization code includes de-serialization code to derive a data object of a class of data objects from a data item of the data exchange format (pg. 2 paragraph [0012] "de-serialization code maps the data of the XML document to the object" "outputs an object that is an instance of a class").

At the time the invention was made, one of ordinary skill in the art would have been motivation to combine the analogous art of using Java and markup

Art Unit: 2124

languages for data exchange of Sundaresan and Tuatini to generate the de-serialization code of Tuatini as a module to derive a data object of the class of data objects from a data item of the data exchange format. The motivation would have been to provide flexibility needed by a group to provide applications tailored to their customers as suggested by Tuatini on pg. 1 paragraph [0004], as the de-serialization code provides a means for this flexibility.

Conclusion

35. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a) Libert et al. (USPN 6,574,655) teaches use of XML.

b) Walsh et al. (USPN 6,810,429) teaches generating a DTD and data mapping utilizing XML.

c) Lee et al. (USPN 6,480,865) teaches transforming documents and adding dynamism to XML documents.

d) Lucas et al. (USPN 6,754,884) teaches mapping data objects of a data representation language to corresponding objects within a programming language and vice versa.

e) Dietrich et al. "A Reusable Graphical User Interface for Manipulating Object-Oriented Databases using Java and XML".

36. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Carlene Gordon whose telephone number is

Art Unit: 2124

(571) 272-3722. The examiner can normally be reached on Mon.-Fri. 8:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

C.G. 1125.

Kakali Chaki
KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.